

Continue to work with your partner on this assignment. Do not forget to switch roles often.

Exercise 3.0: Full House

We are through with manipulating tetrads in the `Tetrad` class. In fact, we are no longer concerned with tetrads themselves but rather with the rows in the `grid` made up of individual blocks. An important task is to clear any rows the user completes (fills full of blocks). The first order of business is to go ahead and complete the `Tetris` private helper method `isCompletedRow`.

```
// precondition: 0 <= row < number of rows
// postcondition: Returns true if every cell in the given row is occupied;
//                false otherwise.
private boolean isCompletedRow(int row)
```

Iterate through each `grid` column in the given row, making a temporary new `Location` object for the row and column in question. Call `grid.get(location)` to see if there is a block there or not. If a given location ever returns `null` (nothing there), then the row is not completed. If you get through the entire row and every location has a block, the row is completed.

Run `TetradTest` to verify that you have completed this exercise before moving on.

Exercise 3.1: Death Row

Next, given a row that is completed, we need to clear that row and then move down (by one row) all of the blocks above that row. Complete the following private helper method `clearRow` in the `Tetris` class. Do not forget to loop through every row above the given row to move down all of the above blocks. Remember that the row numbers will decrease as you move upwards. Also, remember that both the `Block` and its `grid` are keeping track of the `Block`'s location and that it is up to the `Block` class to keep these consistent. In other words, you cannot remove a location from the grid, but rather a given `Block` must `removeSelfFromGrid` instead. *This can be a challenging method to write correctly, so think about what you want to do before you write the code for this method.*

```
// precondition: 0 <= row < number of rows;
//                given row is full of blocks
// postcondition: Every block in the given row has been
//                removed, and every block above row
//                has been moved down one row.
private void clearRow(int row)
```

Run `TetradTest` to verify that you have completed this exercise before moving on.

Exercise 3.2: Clean House

Now use the above helper methods to complete the following `Tetris` method. Remember that you want to start with the bottom row and work your way up to the top. Do not forget that when a row is cleared and everything above moves down one row, you need to recheck it to see if it should also be cleared.

```
// postcondition: All completed rows have been cleared.
private void clearCompletedRows()
```

Whenever a tetrad stops falling, call `clearCompletedRows`. Now play Tetris!

(Yes, your game will misbehave when you lose, but there is a simple workaround to this problem: Do not lose.)

Run `TetradTest` to verify that you have completed this exercise before moving on.

Exercise 3.3: Last Request

Complete the private helper method `topRowsEmpty` in the `Tetris` class.

```
// returns true if top two rows of the grid are empty;  
// false otherwise  
private boolean topRowsEmpty()
```

Use this method to break out of the `play` method when there is no more room to insert a new tetrad at the top of the grid. Finish your game by gradually increasing the speed at which the tetrads fall and calculating and displaying a score in the window title. You may add any enhancements of your own if you wish.

Run `TetradTest` to verify that you have completed this exercise. Please submit a screenshot of the tester success message for Part – 3.

Your Tetris game is now ready to play. Go ahead and run `Tetris`. A random tetrad should appear at the top of your `Tetris` window. You should be able to move it around with the arrow keys, and your code should prevent you from moving the tetrad outside of the window. Pressing the up arrow will rotate the tetrad clockwise. Pressing the space bar will drop it to the floor. When a tetrad hits the floor (or another tetrad), a new tetrad will spawn from the top. The game speeds up with each new tetrad. You will find that the game gets hard quickly. Good luck, and have fun!